Remarks/Arguments

Claims 1-84 are pending in this application.

RESPONSE TO THE EXAMINER'S INTERPRETATIONS OF CLAIM LANGUAGE

The Examiner's interpretation of claim language, element b), is proper as it relates to "the specific request to be made", but is improper as it relates to the length of the timeout period for its particular request." This is because Applicants' dependent claim 3 recites the feature of instantiating a timer, while claim 1 relates only to the specific request to be made, and thus the invention of claim 1 includes both methods that include the instantiation of a timer and that do not include such.

CLAIM REJECTIONS UNDER 35 USC 103

Claims 1-3, 12-18, 27-33, 42-48 and 57-84 are rejected under 35 USC 103(a) as being unpatentable over Nazem et al. (USP 5,983,227) in view of Ferguson (US2002/0178232). Each of claims -3, 12-18, 27-33, 42-48 and 57-84, as amended, is allowable for the reasons that follow. No combination of Nazem et al. and Ferguson teaches or suggests all of the elements any of amended claims 1-3, 12-18, 27-33, 42-48 and 57-84.

Claim 1, as amended, recites that after receiving a single request (specifying multiple content components derived from content hosted by a plurality of distinct component servers), a plurality of information requests for the content are generated as parallel worker threads spawned from a main execution thread. Amended claim 1 further recites sending the plurality of requests as parallel or rapid sequential worker threads so that each information request to the component server hosting the content corresponding to the information request before receiving a response to any of the information requests, thereby permitting concurrent generation of the content components at the component servers. An advantage of Applicants' invention is that the

requested data is retrieved from each of the component servers very soon after a request is received from a client, for example, client 518A of Applicants' Figure 5. The second element of Applicants' claim 1 recites specifically "After receiving the single request, generating a plurality of information requests for the content ...." This feature of Applicants' invention permits the assembly of a network page that has very fresh data.

Applicants' invention advantageously requires generation of multiple requests for multiple content components as parallel worker threads spawned from a main execution thread, thereby permitting concurrent generation of the content components at the component servers. In this way, the data may be retrieved after the request comes in, thereby providing enhanced freshness, while the generation of multiple requests as parallel worker threads permitting concurrent generation of the content components does not result in a delay that is the sum of the time for each retrieval process. Instead, the delay will be only as long as the longest retrieval process for any of the requested components (note that Applicants' invention as set forth at claim 3 sets forth an upper limit on the wait time before generating the personalized page without a component that is delayed beyond a timeout period). The result is that Applicants' invention provides a personalized page with fresher data than any combination of Nazem et al. and Ferguson, and without intolerable delay.

The Examiner concedes that Nazem et al. do not disclose this feature of Applicants' invention. The Examiner relies instead upon paragraph [0121] and Figure 9 of Ferguson. However, this disclosure of Ferguson does not teach or suggest this feature of Applicants' invention. Figure 5 of Ferguson simply illustrates an invention engine 400 that requests a page 330 at a requested Web server 306 which sends the page 332 to the invention engine 400. Figure 6 of Ferguson simply shows a BITE client 408 communicating a request 330 to a Web server 306 through an invention Gateway 402. Figure 8 shows invention Web server 302 receiving outgoing banner request with client ID 712 from Ad Fetcher 708, and sending incoming banner data 714 to Ad Fetcher 708. Figure 9 shows a sequence of communications from top to bottom of Sheet 9 including Ad banner request, then CGI response, then Banner Fetch, then Ad Banner File Delivery, and finally Click Input on Ad Banner.

Paragraph [0121] simply states that an Ad Management System (AMS), such as that illustrated on the left in Figure 9 of Ferguson, works in parallel with a BITE client 408, which is capable of generating HTTP requests to Web servers. There is no mention that a single incoming request includes HTTP requests and Ad requests, and it is doubtful that a user would request an Ad. Instead, a user would send an HTTP request, which would result in Ferguson's BITE client 408 filling the user's HTTP request through invention gateway 402. An Ad, which was not requested in the HTTP request, would be generated by the AMS and combined as a banner in a Web page including the requested content. There is no suggestion that BITE client 408 generates a plurality of information requests for the content requested by the user as parallel worker threads spawned from a main execution thread, and certainly the addition of an Ad banner to requested content does not meet this feature of Applicants' invention.

Applicants' Figures 5-9 and 12-14 illustrate the invention which is clearly distinguished from Ferguson. Applicants' requests involve multiple content components which are provided in response, unlike Ferguson whose requests involve a content component which is provided in combination with a non-requested Ad banner in response.

As no combination of Nazem et al. and Ferguson teaches or suggests to generate multiple requests for multiple requested content components, after receiving the request, as parallel worker threads spawned from a main execution thread, thereby permitting concurrent generation of the content components at the component servers, whereby the retrieving by parallel worker threads greatly reduces delays incurred in serial processing environments such as those described by Nazem et al. and Ferguson, then Applicants' claim 1 is allowable.

Claims 2-3, 12-18, 27-33, 42-48 and 57-84 are allowable for the same reasons as claim 1.

Claims 4-11, 19-26, 34-41 and 49-56 are allowable for the same reasons as claim 1, and because no combination of Nazem et al., Ferguson nor McMichael (US2006/0941339) teaches or suggests all of the elements of Applicants' invention, as set forth at any of these pending claims.

RESPONSE TO THE EXAMINER'S RESPONSE TO REMARKS

Referring to Nazem et al., the Examiner states that "the live data used to fill templates is stored local to the page server". It is respectfully submitted that the locally-stored data is no longer live. Live data would be retrieved directly from Web servers. As is clear in the recitation of Applicants' claim 1, the multiple "live" data content components are retrieved after a request for the data is received. Nazem et al. teaches to store data in local memory 212 from which a content request can be filled rapidly by filling a template with the stored data. Although the request may be quickly filled because the data has already been retrieved from the component servers 230, 232, 234, and stored locally in memory 212, the method of Nazem et al. suffers from a lack of freshness of the data compared with Applicants' invention.

Nazem et al., in fact, teach to assemble a page entirely at the page server from data that has been previously retrieved and stored at the shared memory 212. This requires that the data used must be retrieved from data sources 230, 232, 234 some time before a client request for the web page, contra Applicants' invention.

Nazem et al. clearly do not teach or suggest retrieving the data after receiving a request from a client, e.g., see the Background at column 1, lines 30-47, and column 4, lines 9-23. For example, beginning at column 3, line 65, Nazem et al. state that "[u]sing user templates and a shared memory for the live data, page server 104 can quickly build custom pages in response to a user request. Where the user template is cached, the page can be generated entirely within the page server 104." Beginning at line 10, Nazem et al. further state that "[t]he user will never be faced with a situation where they have to wait for a server to rebuild a page for them by querying the various data providing servers .... Page generator 210 can generate custom front page 218 much more quickly using shared memory 212 as compared with using servers 230, 232, 234 and page template 202. One reason for this is that the time it takes to retrieve data from shared memory 212 does not appreciably increase relative to the bandwidth delay time when more data is retrieved. For example, if stock server 232 were queried for each individual stock quote, a

page with fifty stock quotes might take ten times as long to generate as a page with five stock quotes." And at the Abstract, lines 14-17, Nazem et al. further state that "[w]ith the live data stored in a local, shared memory, any custom page can be built within the page server, eliminating the need to make requests from other servers for portions of the live data."

Nazem et al. teach that it is better to retrieve the data before a client request, then to do so after a request comes in, because this avoids untoward delays due to bandwidth constraints. Nazem et al. contemplate that retrieving the data after a request comes in for a personalized page would result in intolerable delays in a serial processing environment wherein the multiple content components are retrieved consecutively. Where Nazem et al. mention that "a page with fifty stock quotes might take ten times as long to generate as a page with ten stock quotes," they clearly suggest that the delay is equal to the sum of the retrieval processes for each individual content component. Applicants' invention is clearly distinguished from Nazem et al. and Ferguson, as well as any combination thereof, because neither discloses that after receiving a single request specifying multiple content components derived from content hosted by a plurality of distinct component servers, a plurality of information requests for the content are generated as parallel worker threads spawned from a main execution thread.

In view of the above, it is respectfully submitted that the application is now in condition for allowance. The Examiner's reconsideration and further examination are respectfully requested. The Examiner is invited to call the undersigned attorney at 415-828-3244 to discuss any further concerns that he may have.

In view of the foregoing, Applicants respectfully submit that all of the pending claims are allowable, and request the Examiner's early examination of the pending claims in the present application. If the Examiner deems a telephonic discussion would be helpful in the examination of the pending claims, Applicants invite the Examiner to contact the undersigned attorney at (415) 828-3244.

The Commissioner is authorized to charge any deficiencies in fees and credit any overpayment of fees to Deposit Account No. 50-2019. A duplicate page is enclosed.

Respectfully submitted,
JACKSON & CO., LLP

Dated: September 24, 2007

By: /Andrew Vernon Smith/
Andrew V. Smith, Esq.
Reg No. 43,132
Attorneys for Applicant

JACKSON & CO., LLP
6114 La Salle Ave., #507
Oakland, CA 94611-2802

Telephone: 510-652-6418
Facsimile: 510-652-5691

Customer No.: 30349

In view of the foregoing, Applicants respectfully submit that all of the pending claims are allowable, and request the Examiner's early examination of the pending claims in the present application. If the Examiner deems a telephonic discussion would be helpful in the examination of the pending claims, Applicants invite the Examiner to contact the undersigned attorney at (415) 828-3244.

The Commissioner is authorized to charge any deficiencies in fees and credit any overpayment of fees to Deposit Account No. 50-2019. A duplicate page is enclosed.

Respectfully submitted,
JACKSON & CO., LLP

Dated: September 24, 2007

By: /Andrew Vernon Smith/
Andrew V. Smith, Esq.
Reg No. 43,132
Attorneys for Applicant

JACKSON & CO., LLP
6114 La Salle Ave., #507
Oakland, CA 94611-2802

Telephone: 510-652-6418
Facsimile: 510-652-5691

Customer No.: 30349